BR-Explorer: A sound and complete FCA-based retrieval algorithm

Nizar Messai, Marie-Dominique Devignes, Amedeo Napoli, and Malika Smaïl-Tabbone

UMR 7503 LORIA, BP 239, 54506 Vandœuvre-lès-Nancy, FRANCE
{messai,devignes,napoli,smail}@loria.fr
http://www.loria.fr/∼messai

Abstract. In this paper we present the BR-Explorer algorithm, a sound and complete information retrieval algorithm, based on Formal Concept Analysis and domain ontologies. The BR-Explorer algorithm addresses the problem of retrieving the relevant objects for a given query. Initially, a formal context representing the relation between a set of objects and the corresponding set of their attributes is given, and the associated concept lattice is built. The BR-Explorer algorithm starts by generating a formal concept representing the considered query, and classifies this query concept in the concept lattice. Then, the BR-Explorer tries to locate the so-called "pivot" concept in the concept lattice, for building step by step the query result (considering the pivot superconcepts in the concept lattice). Finally, the BR-Explorer algorithm returns a set of relevant data sources that are ranked with respect to their relevance with the query. A query refinement procedure taking advantage of a domain ontology improve the capabilities of the BR-Explorer algorithm, and enrich the result.

1 Introduction

Biological data sources have grown in size and in complexity in the past few years. They are stored under heterogenous formats in distributed localizations. In parallel with the general interest on data mining, the exploitation of this huge amount of data may provide new elements of knowledge, or new hypotheses to be tested in biology. One major problem is to identify relevant data sources according to given criteria. In [9], a method for querying biological data sources with respect to a set of metadata (describing the data sources) and a domain ontology is presented. This method relies on Formal Concept Analysis, and takes advantage of a concept lattice, that can be considered as a kind of indexing structure for organising biological data sources. Moreover, the data sources are described in terms of elements lying in domain ontologies, especially built for biological data sources retrieval (see [12,4,3]).

In this paper, we reuse the biological example introduced in [9], we detail the retrieval procedure, and prove the correctness and completeness of the retrieval algorithm, with respect to a relevance criterion. The retrieval procedure is based

on an algorithm, called BR-Explorer, that is formally described hereafter (section 2). We formally describe and generalize the previous research work presented in [9], showing in that way that it may be generalized to information retrieval based on Formal Concept Analysis principles.

In the following, we suppose that there exist a formal context $\mathbb{K} = (G, M, I)$, where G is a set of objects, M a set of attributes, and I is an incidence relation (on $G \times M$). The set of concepts that may be built from the formal context $\mathbb{K} = (G, M, I)$ is denoted by $\mathfrak{B}(G, M, I)$, and the resulting concept lattice by $\underline{\mathfrak{B}}(G, M, I)$ (see [6]). The BR-Explorer retrieval algorithm is used to satisfy a query of the form $Q = (X, X')$ where X' is a set of given attributes describing the objects to be retrieved.

The paper is written as follows. First, formal definitions are introduced, especially the relevance criterion. Then, the BR-Explorer algorithm is detailed and illustrated by an example in biology (borrowed from [9]). The soundness and the completeness of the BR-Explorer algorithm with respect to the relevance criterion are proved. Finally, a query refinement process, by generalization and by specialization with respect to a domain ontology is proposed, and illustrated with biological examples. Finally, the paper ends with a short discussion and future work.

2 The BR-Explorer retrieval algorithm formalization

2.1 Definitions

Definition 1 (Query). *A query Q is a pair (X,X') where X is a "dummy object" and X' is a set of attributes.*

In the definition 1 the object X is a dummy object supposed to satisfy all the attributes in X'.

As in the most known FCA-based information retrieval algorithms [7,2], BR-Explorer retrieves objects, by classifying a query $Q = (X, X')$ (as stated in definition 1) in a concept lattice indexing the considered objects. The insertion of the query $Q = (X, X')$ in the concept lattice can be considered as the addition of a new entry in the initial formal context. In this way, two alternatives are possible: computing the new concept lattice from scratch or using an incremental classification algorithm such as [8]. The second alternative has been used in the present research work.

Definition 2 ($\oplus$). *For a formal context $\mathbb{K} = (G, M, I)$ and a query $Q = (X, X')$ we define the addition operator $\oplus$ as follow:*

$$\begin{aligned} \mathbb{K}_Q &= \mathbb{K} \oplus Q \\ &= (G, M, I) \oplus (X, X') \\ &= (G \cup X, M \cup X', I_Q) \\ &= (G_Q, M_Q, I_Q) \end{aligned}$$

Definition 3 (Pivot concept). *Consider $\mathbb{K} = (G, M, I)$ a formal context and $Q = (X, X')$ a query. The pivot concept in the concept lattice $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ of the formal context $\mathbb{K}_Q = (G_Q, M_Q, I_Q) = \mathbb{K} \oplus Q$ is the concept $P = (X'', X')$.*

Let us consider $\mathfrak{B}(G, M, I)$ the set of formal concepts of the formal context $\mathbb{K} = (G, M, I)$ and $Q = (X, X')$ a query. According to the relation between X' and the intents of formal concepts in $\mathfrak{B}(G, M, I)$, we distinguish the following cases for the pivot concept $P = (X'', X')$ in the lattice $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$:

- If $\nexists\, C = (A, B) \in \mathfrak{B}(G, M, I)$ such that $X' \subseteq B$ then X and X' are closed respectively in G_Q and M_Q and $X'' = X$. This means that the new entry in the formal context $\mathbb{K} \oplus Q$ yields a new formal concept $(X, X') = (X'', X')$. In addition, each concept $C_1 = (A_1, B_1)$ verifying $B_1 \subseteq X'$ will be transformed into a new concept $C_2 = (A_1 \cup X, B_1)$ in $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$.
- If $\exists\, C = (A, B) \in \mathfrak{B}(G, M, I)$ such that $X' \subseteq B$ then there are two subcases:
  - If $X' \subset B$ then $X'' = A \cup X$ and the pivot concept is $P = (A \cup X, X')$. This means that the new entry in $\mathbb{K} \oplus Q$ will be merged with other entries sharing the same attributes. As in the previous case the operation $\mathbb{K} \oplus Q$ results in creating at least one new formal concept in $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$, namely the pivot concept $P = (A \cup X, X')$. In parallel, each concept $C_1 = (A_1, B_1)$ such that $B_1 \subseteq X'$ is transformed into $C_2 = (A_1 \cup X, B_1)$.
  - If $X' = B$ then $X'' = A \cup X$ and the pivot concept is $P = (A \cup X, X')$. Contrasting the two previous cases, the operation $\mathbb{K} \oplus Q$ does not lead to the creation of any new concept in $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$. It only results in modifying those concepts of the form $C_1 = (A_1, B_1)$ such that $B_1 \subseteq X'$ into $C_2 = (A_1 \cup X, B_1)$.

Definition 4 (upper cover).

(1) Let us consider a formal context $\mathbb{K} = (G, M, I)$, and the associated set of formal concepts $\mathfrak{B}(G, M, I)$ and the concept lattice $\underline{\mathfrak{B}}(G, M, I)$. The upper cover of a formal concept $Y \in \mathfrak{B}(G, M, I)$ is contributed by all the upper neighbors [6] of Y in $\underline{\mathfrak{B}}(G, M, I)$:

$$upper\text{–}cover(Y) = \{C \in \mathfrak{B}(G, M, I) \,|\, Y \leq C \, and \nexists\, Z \in \mathfrak{B}(G, M, I) \,|\, Y \leq Z \leq C\}$$

(2) Given a set $\{C_j\}_{j \in J}$ of formal concepts in $\mathfrak{B}(G, M, I)$, the upper cover of the set $\{C_j\}_{j \in J}$ is defined as the union of upper cover of each concept C_j. We note:

$$upper-cover(\{C_j\}_{j \in J}) = \bigcup_{j \in J} upper-covers(C_j)$$

Definition 5 (Relevance criterion).

(1) Let us consider an entry (a, b) in a formal context $\mathbb{K} = (G, M, I)$, and a query $Q = (X, X')$. the object a is relevant with respect to Q if and only if $b \cap X' \neq \emptyset$, i.e. there is at least one attribute in X' that is associated with the object a.

(2) The degree of relevance of the object a with respect to the query Q is the cardinal of the set $b \cap X'$, i.e. $|b \cap X'|$.

Proposition 1. *Consider a formal context* $\mathbb{K} = (G, M, I)$ *and a query* $Q = (X, X')$*. All the relevant objects with respect to* Q *in* G *are in the extent of the pivot concept* $P = (X'', X')$ *namely* X'' *and the extents of the pivot superconcepts in the concept lattice* $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$*.*

Proof. First let us consider the objects in X'', the extent of the pivot concept. According to the definition of the pivot concept $P = (X'', X')$ (i.e. definition 3) and the definition of relevance (i.e. definition 5), all the objects in X'' are relevant with respect to the query $Q = (X, X')$ since they all share all the attributes in X', the query intent.

Let us now consider the case of the pivot superconcepts. Let $C = (A, B)$ be a superconcept of the pivot concept $P = (X'', X')$ in $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$, i.e. $P = (X'', X') \leq C = (A, B)$. Then, by definition of the lattice ordering, $B \subseteq X'$, meaning that each object in A shares at least an element with X', and hence is relevant.

The most general concept in the lattice, namely $\top$, is not considered when its intent is the empty set since in such case data sources in the extent of this formal concept may not share any metadata with the query.

2.2 The BR-Explorer retrieval algorithm

In this section, we explain the BR-Explorer retrieval algorithm presented hereafter (Algorithm 1). Let us consider a query $Q = (X, X')$ and a formal context $\mathbb{K} = (G, M, I)$ and the associated concept lattice $\underline{\mathfrak{B}}(G, M, I)$.

Intuitively, the BR-Explorer algorithm proceeds as follow. Firstly, the query $Q = (X, X')$ is classified and inserted in the lattice $\underline{\mathfrak{B}}(G, M, I)$ (see Algorithm 1 line 1). This classification yields a new concept lattice $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ and a pivot concept $P = (X'', X')$ (line 2 of the algorithm, where the location of P in $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ is given by the procedure *Locate_Pivot*). The set of objects that are in X'' and in the extents of the superconcepts of P are assigned to the result set $\mathcal{R}_{objects}$ (line 8 and 18 in BR-Explorer algorithm). The rank of each object, i.e. the degree of relevance of the object, is memorized during the object addition to the result. Now let us consider $SUBS_1 = \textit{upper-cover}(P)$. The set of objects in the extents of the concepts in $SUBS_1$ and not already in the result are added to $\mathcal{R}_{objects}$ with the corresponding rank. The next step consists in considering $SUBS_2 = \textit{upper-cover}(SUBS_1)$ the upper neighbors of concepts in $SUBS_1$ and adding new emerging data sources to $\mathcal{R}_{objects}$. Then we continue in the same way for $SUBS_3$, $SUBS_4$ etc until we reach an empty set $SUBS_n$. In each step i, if the concept $\top$ appears in the set of concepts $SUBS_i$ and if the intent of $\top$ is the empty set, then the objects in its extent are ignored since they may not share any attribute with the query.

2.3 Example: biological data sources retrieval

In this section, we detail the application of the BR-Explorer algorithm to the retrieval of biological data sources. Let us consider the formal context $\mathbb{K} =$

Algorithm 1 BR-Explorer algorithm

Require: $\mathbb{K} = (G, M, I)$, $\underline{\mathfrak{B}}(G, M, I)$ and $Q = (X, X')$
Ensure: $R_{sources}$
1: Insert Q into $\underline{\mathfrak{B}}(G, M, I)$
2: $P = (X'', X') := Locate\_Pivot(\underline{\mathfrak{B}}(G_Q, M_Q, I_Q), Q)$
3: $n := 1$ \\$n$ is the level in $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ from P
4: $\mathcal{SUBS}_{n-1} := \{P\}$
5: $rank := 1$
6: **if** $X'' \neq X$ **then**
7: $\mathcal{R}_{rank} := X''\backslash X$
8: $\mathcal{R}_{objects} := (rank, \mathcal{R}_{rank})$
9: $rank := rank + 1$
10: **end if**
11: **while** $\mathcal{SUBS}_{n-1} \neq \emptyset$ **do**
12: $\mathcal{SUBS}_n := covers(\mathcal{SUBS}_{n-1})$
13: $\mathcal{R}_{rank} := \emptyset$
14: **for all** $C = (A, B) \in \mathcal{SUBS}_n$ such that $B \neq \emptyset$ **do**
15: $\mathcal{R}_{rank} := \mathcal{R}_{rank} \cup A$
16: **end for**
17: $EmergingSources := \mathcal{R}_{rank}\backslash(X \cup \mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_{rank-1})$
18: $\mathcal{R}_{objects} := \mathcal{R}sources \cup (rank, EmergingSources)$
19: $n := n + 1$
20: $rank := rank + 1$
21: **end while**

Algorithm 2 Locate_Pivot

Require: $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ and $Q = (X, X')$
Ensure: $P = (X'', X')$
1: $found := false$
2: $\mathcal{SUBS} := \perp$ \\ $\perp$ is the bottom concept in $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$
3: **while** $!found$ **do**
4: **for** *each* $C = (A, B) \in \mathcal{SUBS}$ **do**
5: **if** $X' = B$ **then**
6: $P := C$
7: $found := true$
8: *break*
9: **else if** $X' \subset B$ **then**
10: $\mathcal{SUBS} := covers(\mathcal{SUBS})$
11: *break*
12: **end if**
13: **end for**
14: **end while**

Table 1. The formal context $\mathbb{K} = (G, M, I)$

Sources \ Metadata	Nucleic Sequence	Proteic Sequence	Any Organism	Animals	Vertebrate	Human	Mouse	Manual Revision
Swissprot		x	x					x
RefSeq	x	x	x					x
TIGR-HGI	x					x		
GPCRDB		x	x					x
HUGE	x	x				x		
ENSEMBL	x			x				
Mouse Genome DB		x					x	
Vega Genome Browser		x			x			

(G, M, I) given in table 1. In this example, the objects in G are the biological data sources, and the attributes in M are the metadata describing these data sources. The associated concept lattice $\underline{\mathfrak{B}}(G, M, I)$ is shown on figure 1.

Let us consider the query $Q = (X, X')$, where $X' = \{$*Nucleic Sequences, Human, Manual Revision*$\}$. This query is used to retrieve data sources containing information about nucleic sequences of the human organism, whose data are manually revised (by domain expert). The addition of this query to the formal context $\mathbb{K} = (G, M, I)$ (i.e. classification of Q into $\underline{\mathfrak{B}}(G, M, I)$), $\mathbb{K} \oplus Q$, yields the formal context $\mathbb{K}_Q = (G_Q, M_Q, I_Q)$ shown in table 2.

The concept lattice $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ associated to the formal context $\mathbb{K}_Q = (G_Q, M_Q, I_Q)$ is shown on figure 2.

The steps of the execution of BR-Explorer algorithm in this example are made precise hereafter, and shown on figure 3. The pivot concept returned by the procedure *Locate_Pivot* is the formal concept $P = (X,\{$*Nucleic Sequences, Human, Manual Revision*$\})$. Based on this pivot concept, the BR-Explorer algorithm constructs the result, denoted here $\mathcal{R}_{sources}$ (corresponding to the list $\mathcal{R}_{objects}$), as explained below:

1. In the first step, $\mathcal{SUBS}_0$ contains P, whose extent is X. No data source is added to the result.
2. This step corresponds to the first iteration of the *while loop*. $\mathcal{SUBS}_1$ contains two formal concepts: $(X \cup \{HUGE, TIGR{-}HGI\}, \{Nucleic\,Sequence, Human\})$, and $(X \cup \{RefSeq\}, \{Nucleic\ Sequence, Manual\ Revision\})$. Data sources in the extents of these two concepts are added to the result, in the form of a pair *(rank, set of data sources)*. Here the pair added to $\mathcal{R}_{sources}$ is $(1, \{HUGE, TIGR - HGI, RefSeq\})$.

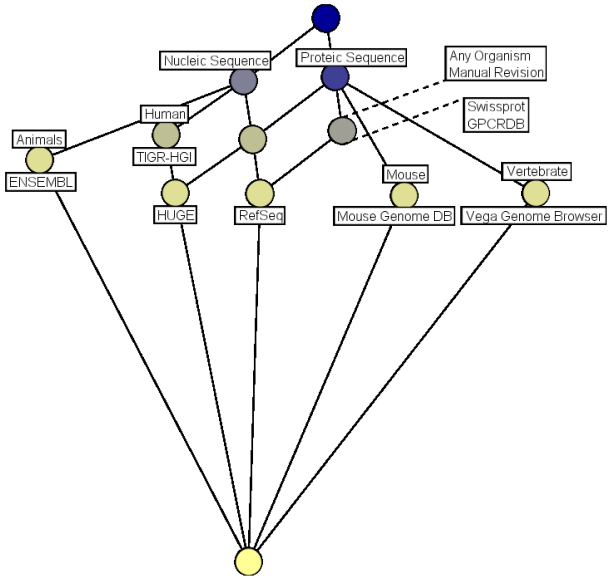

Fig. 1. The concept lattice $\underline{\mathfrak{B}}(G, M, I)$ of the formal context $\mathbb{K} = (G, M, I)$

Table 2. formal context $\mathbb{K}_Q = \mathbb{K} \oplus Q$

Sources \ Metadata	Nucleic Sequence	Proteic Sequence	Any Organism	Animals	Vertebrate	Human	Mouse	Manual Revision
Swissprot		x	x					x
RefSeq	x	x	x					x
TIGR-HGI	x					x		
GPCRDB		x	x					x
HUGE	x	x				x		
ENSEMBL	x			x				
Mouse Genome DB		x					x	
Vega Genome Browser		x			x			
X	x					x		x

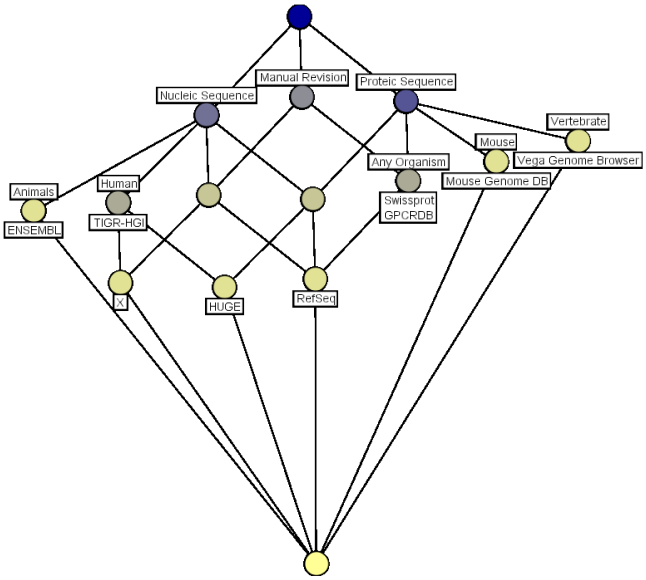

Fig. 2. The concept lattice $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ of the formal context $\mathbb{K}_Q = (G_Q, M_Q, I_Q)$

3. This step corresponds to the second iteration of the *while loop*. $\mathcal{SUBS}_2$ contains two formal concepts: $(X \cup \{HUGE, TIGR-HGI, RefSeq\}, \{Nucleic\ Sequence\})$, and $(X \cup \{RefSeq, Swissprot, GPCRDB\}, \{Manual\ Revision\})$. The emerging data sources in the concept lattice are *Swissprot* and $GPCRDB$, that are added to the result, in the pair (*2*,{*Swissprot,GPCRDB*}).
4. This step corresponds to the third iteration of the *while loop*. $\mathcal{SUBS}_3$ contains only one formal concept, namely the top concept, whose intent is empty. Thus, no data source is added to the result: the *for loop* is skipped, and the set $\mathcal{R}_3$ initialized to $\emptyset$ in line 13 is not modified.
5. This step corresponds to the termination of the algorithm for this example: $\mathcal{SUBS}_4$ does not contain any formal concept.

Finally, the result returned by the BR-Explorer algorithm for the query $Q =$ (*X*,{*Nucleic Sequences, Human, Manual Revision*}) is the following:

$$\begin{aligned}\mathcal{R}_{sources} = \{ & \\ & (1, \{HUGE, TIGR-HGI, RefSeq\}, \\ & (2, \{Swissprot, GPCRDB\}) \\ \} & \end{aligned}$$

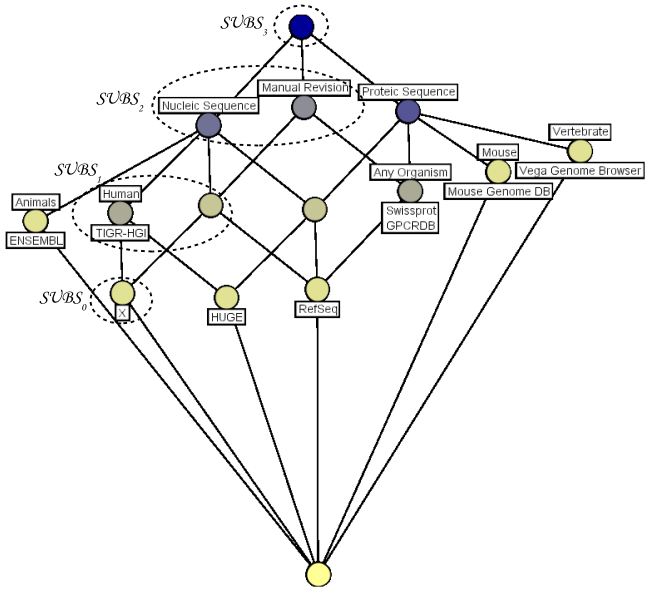

Fig. 3. Steps of the BR-Explorer execution on the concept lattice $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$

3 Soundness and completness

3.1 Soundness

Definition 6. *Given a formal context $\mathbb{K} = (G, M, I)$ and a query $Q = (X, X')$, the BR-Explorer retrieval algorithm is sound with respect to the relevance criterion whenever a retrieved object is relevant for Q.*

Proposition 2. *The BR-Explorer retrieval algorithm is sound with respect to the relevance criterion.*

Proof. Let us consider a query $Q = (X, X')$ and an object g retrieved by BR-Explorer retrieval algorithm. According to the proposition 1, g is retrieved by BR-Explorer as a relevant data source for Q means that g belongs either the extent of the pivot concept or to the extent of a superconcept of the pivot concept. In both cases $\{g\}' \cap X' \neq \emptyset$ proving the the relevance of the object g with respect to the query Q.

3.2 Completeness

Definition 7. *The BR-Explorer retrieval algorithm is complete with respect to the relevance criterion whenever all relevant objects for the considered query in G are retrieved by the algorithm.*

Proposition 3. *The BR-Explorer retrieval algorithm is complete with respect to the relevance criterion.*

Proof. Let us consider a query $Q = (X, X')$ and an object $g \in G$ relevant for Q. Then according to the definition of relevance (i.e. definition 5) $\{g\}' \cap X' \neq \emptyset$. Two cases may be distinguished:

- If $\{g\}' \subset X'$ then $\exists\ C = (A, B) \in \underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ such that $B = \{g\}'$ and $g \in A$, i.e. $C = (\{g\}'', \{g\}')$. This means that C is a superconcept of the pivot concept P in the concept lattice $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$, and that g is in the extent of this superconcept. According to proposition 1, g is retrieved by the BR-Explorer retrieval algorithm.
- If $X' \subseteq \{g\}'$ then $\{g\}'' \subseteq X''$. Based on the fact that $g \in \{g\}''$, then $g \in X''$, i.e. the object g is in the extent of the pivot concept $P = (X'', X')$. According to the proposition 1 g is retrieved by the BR-Explorer retrieval algorithm.

Proposition 2 and proposition 3 allow to state the following theorem.

Theorem 1. *Given a formal context $\mathbb{K} = (G, M, I)$ and a query $Q = (X, X')$, the BR-Explorer retrieval algorithm is sound and complete with respect to the relevance criterion.*

Theorem 1 states that whenever an object $g \in G$ verifies $\{g\}' \cap X' \neq \emptyset$, then g is relevant for query Q, and that every object g verifying $\{g\}' \cap X' \neq \emptyset$ is retrieved by the algorithm.

4 Query refinement

The BR-Explorer retrieval algorithm returns all the relevant objects with respect to the relevance criterion, given a query $Q = (X, X')$ and a formal context $\mathbb{K} = (G, M, I)$. In some cases, it may happen that the result of the retrieval algorithm, namely $\mathcal{R}_{objects}$, is empty, i.e. no object in G fulfills the constraints stated in $Q = (X, X')$. There can be a number of raisons for such a case, e.g. no object in G is described according to the attributes present in X'.

Here the BR-Explorer may take the advantage of domain knowledge, and more precisely, rely on a domain ontology for enhancing an refining the retrieval process. The principle of the refined retrieval process is the following. It is supposed that a domain ontology denoted by $\mathcal{H}_M$ exists, and organizes the attributes in M (given a formal context $\mathbb{K} = (G, M, I)$) in a partial ordering. Roughly speaking (see also [5]) the ontology $\mathcal{H}_M$ is represented as a graph whose vertices correspond to attributes in M, and whose edges correspond to order relations between the attributes in M. In this way, for avoiding to obtain an empty result for the retrieval process, the original query $Q = (X, X')$ enriched with a set of attributes in $\mathcal{H}_M$ related to the existing attributes in X', i.e. X' is transformed into Y' where $Y' \backslash X'$ may be constituted by all the subsumers in $\mathcal{H}_M$ of the attributes present in X'.

For example let us suppose that in the query $Q = (X, X')$, $X' = \{a, b\}$, and that $\mathcal{H}_M$ is represented by the tree shown in figure 4, then $Y' = \{a, b, c, d\}$. Based on $\mathcal{H}_M$, the query may be refined in two ways: either by generalization or by specialization as this is explained hereafter.

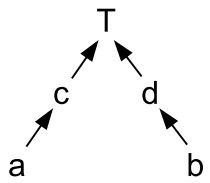

Fig. 4. An example of ontology $\mathcal{H}_M$

4.1 Query refinement by generalization w.r.t. a domain ontology

Let us consider a query $Q = (X, X')$, a domain ontology $\mathcal{H}_M = (V_M, E_M)$, and an attribute $m \in X'$. As an attribute, m belongs the set of vertices of the ontology (actually $V_M = M$). The query refinement process based on generalization consists in adding to X' all attributes subsuming m in $\mathcal{H}_M$. It is supposed, for reasons of completeness, that no other attributes than those present in $\mathcal{H}_M$ are considered, i.e. every object in $\mathbb{K} = (G, M, I)$ is described with and only with attributes in M. In this way, given a query $Q = (X, X')$, and $m \in X'$, the transformation of Q under generalization yields a new query $Q_{gen}(m) = (X, (X' \cup subsumers(m, \mathcal{H}_M) \cap M))$. Then the process of answering the query is the same the retrieval process presented in the preceding sections.

Let us consider the biological example where is given a query $Q = (X, X')$ with $X' = \{Chicken\}$. As no biological source is described using $\{Chicken\}$ (see table 1), the BR-Explorer will return an empty list of biological sources. Relying on the *BioR ontology* (see figure 5), the ancestors of $\{Chicken\}$ that are in M: *Vertebrates, Animals*, and *Any organism*, the new query becomes: $Q_{gen}(m) =$ (*X*,{ *Vertebrates, Animals, Any organism*}). This time the BR-Explorer retrieval algorithm will return the following list of biological sources:

$$\begin{aligned}\mathcal{R}_{sources} = (1, \{&\text{ENSEMBL,}\\ &\text{Vega Genome Browser,}\\ &\text{Swissprot,}\\ &\text{RefSeq,}\\ &\text{GPCRDB}\}).\end{aligned}$$

4.2 Query refinement by specialization w.r.t. a domain ontology

Given a query $Q = (X, X')$, an attribute $m \in X'$ and a domain ontology $\mathcal{H}_M$, the query refinement by specialization is defined in a dual way with respect to

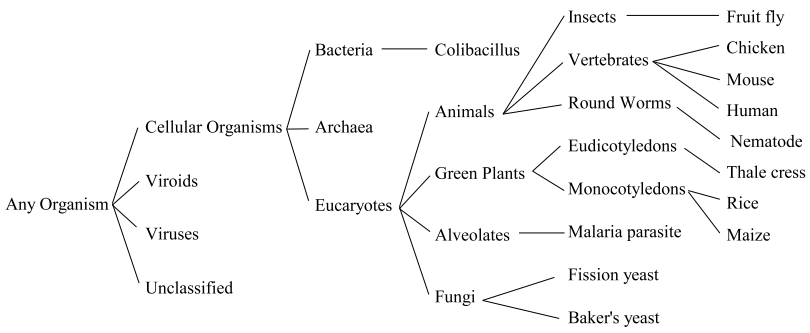

Fig. 5. The BioR ontology

the query refinement by generalization. Given $m \in X'$, let us consider the descendants (*subsumees*) of m in $\mathcal{H}_M$ that are also in M, i.e. $subsumees(m, \mathcal{H}_M)$. Then the new query is defined as $Q_{spe}(m) = (X, (X' \cup subsumees(m, \mathcal{H}_M)) \cap M)$. Then, the retrieval process can be performed in the standard way.

Let us return to our previous example (section 2.3), and consider the query $Q = (X, \{Eucaryotes\})$. According to the *BioR ontology*, the query Q is transformed as $Q_{spe}(m) = (X, \{Animals,\ Vertebrate,\ Human,\ Mouse\})$. The BR-Explorer retrieval algorithm may then be applied to return the following list of biological data sources:

$$\begin{aligned}\mathcal{R}_{sources} = (1, \{&\text{ENSEMBL},\\ &\text{Vega Genome Browser},\\ &\text{Swissprot},\\ &\text{Mouse Genome DB}\}).\end{aligned}$$

4.3 The contribution of the ontology-based query refinement

The contribution of the query refinement based on the attribute ontology may be illustrated by the proposition 4. In the following, $Q_R = (X_R, X'_R)$ denotes the refined query based on $Q = (X, X')$, where $X_R = X$ and $X'_R = X' \cup M_R$, M_R being the set of attributes added within the refinement process. The refined pivot concept becomes $P_R = (X''_R, X'_R)$.

Proposition 4. *Let us consider a query $Q = (X, X')$, an ontology $\mathcal{H}_M$, and the corresponding refined query $Q_R = (X_R, X'_R)$. The list of objects returned by the BR-Explorer retrieval algorithm answering the $Q_R = (X_R, X'_R)$ includes as a subset the list of objects answering the query $Q = (X, X')$.*

Proof. Let us consider $P = (X'', X')$ and $P_R = (X''_R, X'_R)$, where $X'_R = X' \cup M_R$. We have $X' \subseteq X'_R$ showing that P_R is a concept more specific than P in the concept lattice $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$. This means that the set of concepts that may be explored starting from $P_R = (X''_R, X'_R)$ includes the set of concepts starting from

$P = (X'', X')$. Accordingly, the list of objects resulting from P_R will include the list of objects resulting from P.

For example, let us consider the examples of query refinement proposed in sections 4.1 and 4.2. In the first case, the query $Q = (X, \{Chicken\})$ yields an empty result. It is refined by generalization into the query $Q_R = (X, \{$*Vertebrates, Animals, Any Organism*$\})$ that yields the result:

$$\mathcal{R}_{sources} = (1, \{\text{ENSEMBL, Vega Genome Browser, Swissprot, RefSeq, GPCRDB}\}).$$

In the second case, the query $Q = (X, \{Eucaryotes\})$ yields an empty result. It is refined by specialization into the query $Q_R = (X, \{$*Animals, Vertebrate, Human, Mouse*$\})$ that yields the result:

$$\mathcal{R}_{sources} = (1, \{\text{ENSEMBL, Vega Genome Browser, Swissprot, Mouse Genome DB}\}).$$

Both types of query refinement may be used independently or conjointly. In the last case, the two types of refinement may provide a more complete and more precise result. There is a discussion on the subject in [9]. Moreover, the number of attributes that are added during the refinement process may be controlled, for example by giving a weight to the attributes, by limiting the number of attributes, i.e. picking only the nearest subsumers or subsumees in $\mathcal{H}_M$, etc. This kind of investigation and its effects on the query result has still to be carried on.

4.4 Relevance of the objects returned by the query refinement

The relevance of the objects that are returned after the query refinement process arises from the fact that they appear in the result list thanks to the addition of one or more new attributes. The new attributes are provided by the domain ontology, and they are semantically related, i.e. by specialization or generalization, with one of the attributes lying in the original query. Thus, there is no way to introduce, in the result of the query, an irrelevant object, i.e. an object that does not share any attribute with the query. However, the degree of relevance of the added objects may be low. For example, for some attribute, say m_1, related to an attribute in the query, say m, in the ontology, it may happen that the distance between m_1 and m is high, i.e. in terms of the edge number in the domain ontology. Then the relevance of the objects resulting from the addition of m_1 may be low: in case of refinement by generalization, the returned objects may be too general and less informative, and in case of refinement by specialization, the returned objects may be too specific and focus on a very narrowed topic for being of interest regarding the original query.

5 Conclusion and further work

The BR-Explorer algorithm presented in this paper is aimed at information retrieval and query answering in a concept lattice. It relies on formal concept analysis and domain ontologies, and it has been successfully applied in biology [9]. The completeness and the soundness of the BR-Explorer algorithm, with respect to a relevance criterion, have been proved. This guarantees that, given a query, the objects that are returned by the algorithm, satisfy the constraints associated to the query. One original aspect characterizing the BR-Explorer algorithm is the way objects retrieved and the way the result is built. This aspect is dependent on the relevance criterion, and on a query refinement, by generalization and specialization, taking advantage of a domain ontology. This gives to the BR-Explorer algorithm a different behavior, contrasting other information retrieval approaches in the field of formal concept analysis, such as those presented in [1], [10] and [11] (some details about these differences are proposed in [9]).

The BR-Explorer algorithm has been successfully applied to retrieve relevant biological data sources according to given constraints. Moreover, the BR-Explorer algorithm can be used in other application domains, that can be formalized using a set of objects and a set of corresponding attributes, and where queries can be considered as a pair $Q = (X, X')$, as stated above.

In the future, we plan to consider a number of research topics, including the introduction of preferences and weights associated with the attributes, i.e. for controlling and making the retrieval more precise, nested queries and the definition of a general query language, i.e. for having the power of query language such as SQL, global reasoning on queries, i.e. for satisfying queries by analogy for example, and finally complex formal contexts, i.e. multi-valued and fuzzy formal contexts.

Acknowledgments

We would like to thank Sergei O. Kuznetsov for his enriching discussion on the BR-Explorer algorithm. This work was supported by the "PRST Intelligence Logicielle" from the Région Lorraine.

References

1. Claudio Carpineto and Giovanni Romano. A lattice conceptual clustering system and its application to browsing retrieval. *Machine Learning*, 24(2):95–122, August 1996.
2. Claudio Carpineto and Giovanni Romano. Order-theoretical ranking. *Journal of the American Society for Information Science*, 51(7):587–601, May 2000.
3. Marie-Dominique Devignes, Nizar Messai, Amedeo Napoli, Shazia Osman, and Malika Smail-Tabbone. Intelligent access to genomic sources on the web. In *W3C Workshop on Semantic Web and Life Sciences (position paper)*, Cambridge, MA, USA, October 2004.

4. Marie-Dominique Devignes, Malika Smail, and Nacer Boudjlida. Collecte de donnés biologiques à partir de sources multiples et hétérogènes. vers une structure de médiation conviviale et orientée source. In *Journées scientifiques sur le Web sémantique*, Paris, Octobre 2002.
5. Dieter Fensel, James A. Hendler, Henry Lieberman, and Wolfgang Wahlster, editors. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2003.
6. Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis*. Springer, mathematical foundations edition, 1999.
7. Robert Godin, Guy W. Mineau, and Rokia Missaoui. Méthodes de classification conceptuelle basées sur les treillis de Galois et applications. *Revue d'intelligence artificielle*, 9(2):105–137, 1995.
8. Robert Godin, Rokia Missaoui, and Hassan Alaoui. Incremental Concept Formation Algorithms Based on Galois (Concept) Lattices. *Computational Intelligence*, 11:246–267, 1995.
9. Nizar Messai, Marie-Dominique Devignes, Amedeo Napoli, and Malika Smail-Tabbone. Querying a bioinformatic data sources registry with concept lattices. In G. Stumme F. Dau, M.-L. Mugnier, editor, *ICCS*, Lecture Notes in Computer Science, pages 323–336. Springer-Verlag Berlin Heidelberg, 2005.
10. Uta Priss. Lattice-based Information Retrieval. *Knowledge Organization*, 27(3):132–142, 2000.
11. Brigitte Safar, Hassen Kefi, and Chantal Reynaud. OntoRefiner, a user query refinement interface usable for Semantic Web Portals. In *Proceedings of Application of Semantic Web technologies to Web Communities, Workshop ECAI'04*, pages 65–79, Valencia, Spain, August 2004.
12. Malika Smail-Tabbone, Shazia Osman, Nizar Messai, Amedeo Napoli, and Marie-Dominique Devignes. Bioregistry: a structured metadata repository for bioinformatic databases. In *The 1st International Symposium on Computational Life Science, CompLife'05, Konstanz, Germany, September 25-27, 2005*, Konstanz, Germany, September 2005.